REMARKS/ARGUMENTS

In the Office Action, the Examiner noted that claims 1-38 are pending in the application. The Examiner additionally stated that claims 1-38 are rejected. By this amendment, claims 10 and 18 have been cancelled and claims 1, 19-20, 24, 27 and 38 have been amended. Hence, claims 1-9, 11-17 and 19-38 are pending in the application.

Applicant hereby requests further examination and reconsideration of the application, in view of the foregoing amendments.

In the Claims

Rejections Under 35 U.S.C. §112, second paragraph

The Examiner rejected claim 19 under 35 U.S.C. 112 second paragraph as being indefinite. Applicant has amended the claim to clarify that the computer-readable program code provides the cache memory recited in claim 1.

Rejections Under 35 U.S.C. §102(b)

The Examiner rejected claims 1-6, 8, 15-21, 23, 26-35 and 38 under 35 U.S.C. 102(b) as being anticipated by Matthews, U.S. Patent No. 5,956,752 (hereinafter, *Matthews*). Applicant respectfully traverses the Examiner's rejections.

Matthews is directed to an index prediction apparatus for accessing data in a cache memory array. *Matthews* solves the problem introduced in a virtual memory system in which the page size is smaller than the amount of data stored in a single way of a set associative cache. In such a virtual memory system, the number of address bits required to index the cache memory array is greater than the number of untranslated address bits. *Matthews* refers to the required address bits beyond the untranslated address bits as the relevant upper portion of the virtual address. To avoid undesirably having to wait for the relevant upper portion of the virtual address to be translated by a TLB in order to index the memory array, *Matthews* predicts the bits of the relevant upper portion of the virtual address with translated bits saved from a previous access of the array. While *Matthews* is indexing the array using the predicted bits, he translates the relevant upper portion of the virtual address and then compares the translated bits with the predicted bits. If the

comparison reveals a mismatch, then *Matthews* re-accesses the array using the translated bits in order to obtain the correct data.

Matthews stores the predicted bits in a latch. In one embodiment, *Matthews* maintains a latch for loads and a separate latch for stores and uses the appropriate prediction bits based on whether the current array access is a load or store. This improves the accuracy of his prediction. *Matthews* also briefly describes an embodiment in which the single load latch is replaced with multiple latches for multiple types of loads, e.g., floating point loads, integer loads, loads of a particular size, and stack loads (pops). Conversely, the single store latch may be replaced with multiple latches for multiple types of stores, e.g., floating point stores, integer stores, stores of a particular size, and stack stores (pushes). In the embodiment in which *Matthews* saves prediction bits for pops and pushes distinct from other loads and stores, the memory array is employed to store data both from pushes and from other types of stores.

With respect to claim 1, Applicant has amended the claim to clarify that the recited LIFO memory is configured to store only data specified by push instructions. Applicant respectfully asserts that *Matthews* does not teach a LIFO memory that stores only data specified by push instructions.

Further with respect to claim 1, the Examiner states that *Matthews* teaches a comparator for comparing a virtual address of a cache line stored in a top storage element of the LIFO memory with a source virtual address of data requested by a load instruction, citing comparator 140 of Fig. 5 and col. 5, lines 28-30. Applicant respectfully asserts that col. 5, lines 28-30 and col. 6, lines 23-26 of *Matthews* teach that comparator 140 compares the upper relevant portion of the predicted virtual address 124 of Fig. 5 with the relevant upper portion of the physical address 122 of Fig. 5, which is the upper relevant portion of the load instruction virtual address 100 of Fig. 5 translated by the TLB 110 of Fig. 5. Thus, the comparator 140 does not compare a virtual address of a cache line stored in a top storage element of the LIFO memory with a source virtual address of data requested by a load instruction, as recited by claim 1. First, comparator 140 compares (the relevant upper portion of) a physical address and (the relevant upper portion of) a predicted

address, which is either (the relevant upper portion of) the load instruction virtual address or (the relevant upper portion of) the address provided by the latches; it does not compare two virtual addresses, as recited in claim 1. Second, comparator 140 does not compare the address of a cache line stored in the memory, as recited in claim 1; rather, it compares (the relevant upper portion of) the physical address translated from (the relevant upper portion of) the virtual address specified by the load instruction with either (the relevant upper portion of) the predicted address stored in the latches or (the relevant upper portion of) the virtual address specified by the load instruction.

Still further with respect to claim 1, the Examiner states that *Matthews* teaches an output for indicating the data requested by the load instruction is present in the cache memory if the comparator indicates the source virtual address matches the cache line virtual address stored in the top storage element, citing prediction wrong signal 178 of Fig. 5 and col. 5, lines 28-36. Applicant respectfully asserts that col. 5, lines 28-36 and col. 6, lines 10-22 teach that prediction wrong output 178 indicates that (the relevant upper portion of) the predicted address 124 does not match (the relevant upper portion of) the physical address 122. Thus, the output 178 does not indicate the data requested by the load instruction is speculatively present in the cache memory. Rather, output 178 simply indicates that *Matthews*' data cache incorrectly predicted the relevant upper portion of the index into his memory array; however, output 178 does not indicate whether the requested data is present in the cache memory.

Finally with respect to claim 1, the Examiner states that it is apparent from the specification that *Matthews*' memory array stores both a virtual and physical address because the memory array can be accessed by both virtual and physical address. Applicant respectfully asserts that this is not apparent, nor is it inferable from *Matthews*' specification. Applicant can find no teaching in the specification of *Matthews* that the memory array stores both virtual and physical addresses. Col. 3, lines 49-53 of *Matthews* teaches that the cache includes a tag array that holds addresses of the data stored in the memory array, so that a check can be performed to determine whether the cache has a requested data item. However, there is no teaching that both a virtual and physical address for each data item are stored in the tag array, nor can Applicant find any other

teaching in *Matthews* from which to infer that both a virtual and physical address for each data item are stored in the tag array. Rather, the strong inference to be made from a fair reading of *Matthews* is that his tag array stores only a physical address tag for each cache line just as conventional caches do. This strong inference may be made from the fact that *Matthews* does not teach that the tag array is different from conventional tag arrays, and in particular, *Matthews* does not discuss how his cache determines whether the requested data is present in the cache, i.e., whether the request hits or misses in the cache. Rather, *Matthews* focuses on his point of novelty, which is predicting the relevant upper portion of the load/store address in order to index his tag and memory array without first translating the virtual load/store address into a physical load/store address. Thus, the reader must conclude that *Matthews* determines whether the requested data is present in his cache by storing a physical address tag in his tag array and comparing the physical address tags for each way in the selected set with the translated physical address 122 to determine whether a cache hit occurred. In summary, because *Matthews* does not explicitly teach that he stores a virtual address in his tag array and does not include any reason to do so, the strongest inference is that *Matthews* does not store a virtual address in his tag array.

For the foregoing reasons, Applicant respectfully submits that *Matthews* does not anticipate claim 1 and respectfully requests the Examiner to withdraw the rejection. Furthermore, because claims 2-9, 11-17 and 19 depend from claim 1 and add further limitations, Applicant respectfully requests the Examiner to withdraw the rejections to those claims.

With respect to claim 20, Applicant has amended the claim to recite the limitations that the first and second cache memories are level-one cache memories. Applicant respectfully asserts that *Matthews* does not teach two level-one cache memories. The Examiner cited external cache 20 of Fig. 1 of *Matthews* as corresponding to the recited second cache memory. *Matthews* teaches at col. 3, lines 43-46 that the external cache 20 is not a level-one cache. Furthermore, claim 20 recites that the microprocessor comprises the first and second cache memories. Applicant respectfully points out that the processor 10 of *Matthews* does not comprise the external cache 20 of Fig. 1.

Further with respect to claim 20, Applicant has amended the claim to recite the limitation that the first cache memory caches data specified only by push instructions and the limitation that the second cache memory caches data specified only by non-push instructions. *Matthews* does not teach that his cache 12 of Fig. 5 caches data specified only by push instructions, and *Matthews* does not teach that his external cache 20 caches data specified only by non-push instructions.

Finally, with respect to claim 20, Applicant respectfully asserts that *Matthews* does not teach his cache memory speculatively providing from a top entry thereof data specified by a load instruction if a virtual address specified by the load instruction matches a virtual address of the cache line stored in the top entry, as recited by claim 20, as discussed above with respect to claim 1.

For the foregoing reasons, Applicant respectfully submits that *Matthews* does not anticipate claim 20 and respectfully requests the Examiner to withdraw the rejection. Furthermore, because claims 21-26 depend from claim 20 and add further limitations, Applicant respectfully requests the Examiner to withdraw the rejections to those claims.

With respect to claim 27, Applicant respectfully asserts that neither col. 2, lines 35-43 nor any other portion of *Matthews* teaches determining whether a virtual address of a load instruction matches a virtual address of data stored in an entry of the cache memory, as recited in claim 27 and as discussed above with respect to claim 1, much less the top entry, and much less before determining whether the physical address of the load instruction matches the physical address of the data stored in the top entry. Therefore, Applicant respectfully submits that *Matthews* does not anticipate claim 27 and respectfully requests the Examiner to withdraw the rejection. Furthermore, because claims 28-37 depend from claim 27 and add further limitations, Applicant respectfully requests the Examiner to withdraw the rejections to those claims.

With respect to claim 38, Applicant has amended the claim to recite the limitations with which claim 1 has been amended. Thus, claim 38 as amended recites a computer program product embodied on a computer-readable medium having computer-readable program code for providing the cache memory recited in amended claim 1. Therefore,

for the reasons stated above with respect to claim 1, Applicant respectfully submits that *Matthews* does not anticipate claim 38 and respectfully requests the Examiner to withdraw the rejection.

Rejections Under 35 U.S.C. §103(a)

The Examiner rejected claim 7 under 35 U.S.C. 103(a) as being unpatentable over *Matthews* in view of Krolak et al., U.S. Patent No. 5,751,990. The Examiner rejected claims 9-14, 22, 24-25 and 36-37 under 35 U.S.C. 103(a) as being unpatentable over *Matthews* in view of Lynch, U.S. Patent No. 5,930,820. Applicant respectfully traverses the Examiner's rejections.

Claims 7 and 9-14 depend from independent claim 1, claims 22 and 24-25 depend from independent claim 20, and claims 36-37 depend from independent claim 27, respectively, which are not anticipated by *Matthews* as discussed above, and recite further limitations. Applicant, therefore, respectfully requests the Examiner withdraw his rejection of these claims.

CONCLUSIONS

In view of the arguments advance above, Applicant respectfully submits that claims 1-9, 11-17 and 19-38 are in condition for allowance. Reconsideration of the rejections is requested, and allowance of the claims is solicited.

Applicant earnestly requests that the Examiner contact the undersigned practitioner by telephone if the Examiner has any questions or suggestions concerning this amendment, the application, or allowance of any claims thereof.

Respectfully submitted,

/E. Alan Davis/

By:_______________________________

E. ALAN DAVIS
Registration No. 39,954
Tel: (512) 301-7234

6/19/2006

Date: _______________________________